

#2

Docket No. 1095.1184

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
	)	
Katsumi ICHINOSE, et al.	)	
	)	Group Art Unit: Unassigned
Serial No.: NEW	)	
	)	Examiner: Unassigned
Filed: April 19, 2001	)	
	)	
For: INFORMATION PROCESSING	)	
METHOD AND RECORDING	)	
MEDIUM	)	

J1017 U.S. PTO  
09/838166  
04/20/01

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Honorable Commissioner of Patents and Trademarks  
Washington, D.C. 20231*

*Sir:*

In accordance with the provisions of 37 C.F.R. §1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-360397, filed: November 28, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,  
STAAS & HALSEY LLP

Date: April 19, 2001

By: \_\_\_\_\_

James D. Halsey, Jr.  
Registration No. 22,729

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001  
(202) 434-1500

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

J1017 U.S. PTO  
09/838166  
04/20/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年11月28日

出 願 番 号

Application Number:

特願2000-360397

出 願 人

Applicant (s):

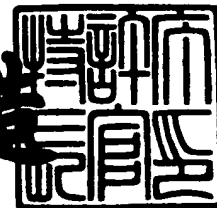
富士通株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 2月23日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



出証番号 出証特2001-3011344

【書類名】 特許願

【整理番号】 0050916

【提出日】 平成12年11月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明の名称】 情報処理方法および記録媒体

【請求項の数】 5

【発明者】

    【住所又は居所】 静岡県静岡市南町18番1号 株式会社富士通静岡エンジニアリング内

    【氏名】 一瀬 克己

【発明者】

    【住所又は居所】 静岡県静岡市南町18番1号 株式会社富士通静岡エンジニアリング内

    【氏名】 守屋 勝由

【特許出願人】

    【識別番号】 000005223

    【氏名又は名称】 富士通株式会社

【代理人】

    【識別番号】 100092152

    【弁理士】

    【氏名又は名称】 服部 毅巖

    【電話番号】 0426-45-6644

【手数料の表示】

    【予納台帳番号】 009874

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

【物件名】            要約書    1

【包括委任状番号】   9705176

【プルーフの要否】    要

【書類名】 明細書

【発明の名称】 情報処理方法および記録媒体

【特許請求の範囲】

【請求項 1】 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させる情報処理方法において、

実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割ステップと、

前記並列処理ブロック分割ステップによって分割された並列処理ブロックを、前記複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割ステップと、

所定のプロセッサにおいて、前記スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示ステップと、

を有することを特徴とする情報処理方法。

【請求項 2】 前記指示ステップは、前記実行対象となるプログラムに所定の指示がなされている場合には、全てのスレッドの処理が終了するまで、次の並列処理ブロックの実行を指示しないことを特徴とする請求項 1 記載の情報処理方法。

【請求項 3】 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

コンピュータを、

実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割手段、

前記並列処理ブロック分割手段によって分割された並列処理ブロックを、前記複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割手段、

所定のプロセッサにおいて、前記スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示手段、

として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体

【請求項4】 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

コンピュータを、

複数の並列処理ブロックに分割された実行対象のプログラムの所定の並列処理ブロックから実行要求がなされた場合には、複数のスレッドを生成し、各プロセッサに処理を分担させる処理分担手段、

何れかのスレッドの処理が終了した場合には、次の並列処理ブロックに係るスレッドの実行を指示する実行指示手段、

として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体

【請求項5】 前記実行指示手段は、前記実行対象となるプログラムに所定の指示がなされている場合には、全てのスレッドの処理が終了するまで、次の並列処理ブロックの実行を指示しないことを特徴とする請求項4記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は情報処理方法および記録媒体に関し、特に複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させる情報処理方法および記録媒体に関する。

【0002】

【従来の技術】

複数のプロセッサを有する計算装置にプログラムを実行させる場合には、対象となるプログラムを複数の並列処理ブロックに分割し、得られた並列処理ブロックを、処理の基本単位であるスレッドに分割して各プロセッサに分担して処理させる方法が一般的であった。

【0003】

図10は、このような方法により図の左側に示すプログラムを5つのプロセッ

サを有する計算装置で実行する場合の様子を示す図である。この例では、実行対象となるプログラムには2つの処理ループ（行番号「1」～「3」および行番号「5」～「7」）が含まれており、それぞれが第1の並列処理ブロックおよび第2の並列処理ブロックとされている。

#### 【0004】

このプログラムが実行されると、各プロセッサに対応する処理の基本単位であるスレッド#1～#5が生成され、スレッド#1は、最初の処理ループ（行番号「1」～「3」のループ）の変数*i*が1から200までの処理を分担し、スレッド#2は、同じく最初の処理ループの変数*i*が201から400までの処理を分担する。同様にして、スレッド#3、スレッド#4、および、スレッド#5は、それぞれ、401から600、601から800、および、801から1000までの処理を分担する。

#### 【0005】

##### 【発明が解決しようとする課題】

しかし、従来においては、並列処理ブロック間には「バリア」と呼ばれる監視機構が設けられており、全てのスレッドの処理が終了するまで次の並列処理ブロックの実行が保留されていた。

#### 【0006】

従って、複数の並列処理ブロックが存在する場合には、処理に要する時間は各並列処理ブロックにおいて最も遅いスレッドの処理時間を加算した時間となり、プロセッサ資源が有効に活用できない場合があるという問題点があった。

#### 【0007】

本発明は、以上のような点に鑑みてなされたものであり、プロセッサ資源を有効活用することにより、処理時間を短縮することが可能な情報処理方法を提供することを目的とする。

#### 【0008】

##### 【課題を解決するための手段】

本発明では上記課題を解決するために、図1に示す、複数のプロセッサ（プロセッサ群4）を有する計算装置に対して、所定の情報処理を実行させる情報処理

方法において、実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割ステップ1と、並列処理ブロック分割ステップ1によって分割された並列処理ブロックを、複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割ステップ2と、所定のプロセッサにおいて、スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示ステップ3と、を有することを特徴とする情報処理方法が提供される。

#### 【0009】

ここで、並列処理ブロック分割ステップ1は、実行対象となるプログラムを複数の並列処理ブロックに分割する。スレッド分割ステップ2は、並列処理ブロック分割ステップ1によって分割された並列処理ブロックを、複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割する。指示ステップ3は、所定のプロセッサにおいて、スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する。

#### 【0010】

##### 【発明の実施の形態】

以下、本発明の実施の形態を図面を参照して説明する。

図1は、本発明の情報処理方法の原理を説明する原理図である。この図に示すように、本発明に係る情報処理方法は、プロセッサ群4を有する情報処理装置に所定の情報処理を実行させることを目的としている。ここで、本発明に係る情報処理方法は、並列処理ブロック分割ステップ1、スレッド分割ステップ2、および、指示ステップを有している。

#### 【0011】

並列処理ブロック分割ステップ1は、実行対象となるプログラムを複数の並列処理ブロックに分割する。

スレッド分割ステップ2は、並列処理ブロック分割ステップ1によって分割された並列処理ブロックを、プロセッサ群4のそれぞれのプロセッサに分担して処理させるための基本処理単位であるスレッドに分割する。

#### 【0012】



指示ステップ3は、所定のプロセッサにおいて、スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示するが、プログラムに所定の指示がなされている場合には、全てのスレッドの処理が終了するまで、次の並列処理ブロックの実行を指示しない。

【0013】

次に、以上の原理図の動作について説明する。

いま、実行対象のプログラムが入力されると、並列処理ブロック分割ステップ1は、これを複数の並列処理ブロックに分割する。なお、並列処理ブロックとは、例えば、ループ処理などのような一定の機能的な一体性を有する処理単位をいう。従って、プログラムに複数のループ処理が含まれている場合には、これらがそれぞれ並列処理ブロックとして分割される。

【0014】

スレッド分割ステップ2は、並列処理ブロック分割ステップ1によって生成された並列処理ブロックのそれぞれを実行される順に取得し、各プロセッサに分担させるべき処理単位としてのスレッドに分割する。図1の例では、5つのプロセッサが存在しているので、並列処理ブロックは5つのスレッドに分割される。

【0015】

プロセッサ群4は、スレッド分割ステップ2によって生成されたスレッドを、各プロセッサによって分担して処理する。このとき、プロセッサによって実行される処理の内容は、同一ではないので、処理が終了するタイミングは、プロセッサ毎に異なることになる。

【0016】

従来においては、全てのプロセッサによる処理が完了するまで、バリアが監視していたので、あるプロセッサが先に処理を終了した場合であっても他の全てのプロセッサが処理を完了するまで待つ必要があった。

【0017】

しかしながら、本実施の形態では、所定のプロセッサにおいて処理が完了した場合には、プログラムに指示がない限り、指示ステップ3が次の並列処理ブロックの実行を指示するので、先にスレッドの実行を終えたプロセッサは次の並列処

理ブロックのスレッドを実行することになる。

【 0 0 1 8 】

そして、実行処理が継続し、所定の指示がプログラムに出現した場合には、指示ステップ 3 が次の並列処理ブロックへの移行を保留するので、そこで、全てのスレッドが同期することになる。

【 0 0 1 9 】

以上に説明したように、従来では各並列処理ブロックにおける最長のスレッドの実行時間を加算した値が実行時間であったが、本発明の情報処理方法によれば、所定の並列ブロックに全ての最長のスレッドが分担されない限りは、実行時間を短縮することが可能となる。

【 0 0 2 0 】

次に、本発明の実施の形態について説明する。

図 2 は、本発明の情報処理方法を実行する実施の形態の構成例を示す図である。

【 0 0 2 1 】

本発明の情報処理方法は、図 1 に示すような情報処理装置において実行される。ここで、情報処理装置 1 0 は、PU (Processor Unit) 1 0 a - 1 ~ 1 0 a - 5、ROM (Read Only Memory) 1 0 b、RAM (Random Access Memory) 1 0 c、HDD (Hard Disk Drive) 1 0 d、GB (Graphics Board) 1 0 e、I / F (Interface) 1 0 f、および、バス 1 0 g によって構成されており、その外部には表示装置 1 1 および入力装置 1 2 が接続されている。

【 0 0 2 2 】

ここで、PU 1 0 a - 1 ~ 1 0 a - 5 は、HDD 1 0 d に格納されたプログラムに従って、各種演算処理を実行するとともに、装置の各部を制御する。

ROM 1 0 b は、PU 1 0 a - 1 ~ 1 0 a - 5 が実行する基本的なプログラムやデータ等を格納している。

【 0 0 2 3 】

RAM 1 0 c は、PU 1 0 a - 1 ~ 1 0 a - 5 が実行対象とするプログラムや演算途中のデータを一時的に格納する。

HDD10dは、PU10a-1～10a-5が実行するプログラムやデータを格納している。具体的には、システムを管理し、基本的なユーザ操作環境を提供するための基本的なプログラムであるOS (Operating System) や、本発明に係るコンパイラ、リンカ、および、実行対象となるアプリケーションプログラム等を格納している。

【0024】

GB10eは、PU10a-1～10a-5から供給された描画命令に従って描画処理を施し、得られた画像を映像信号に変換して出力する。

I/F10fは、入力装置12から出力されたデータを、装置内部の表現形式に変換して入力する。

【0025】

バス10gは、PU10a-1～10a-5、ROM10b、RAM10c、HDD10d、GB10e、および、I/F10fを相互に接続し、これらの間で情報の授受を可能とする。

【0026】

なお、PU10a-1～PU10a-5は、並行して処理を行うことが可能であり、実行対象のプログラムを複数の並列処理ブロックに分割して処理する。その際、共通のリソースであるROM10b等にアクセスする際には、排他制御により、他のユニットの影響を受けないように制御されている。

【0027】

また、以上の構成例は、説明を一部簡略化して示してあるが、要は複数のPUと、それぞれのPUが独立して動作するような構成を有していればよい。

図3は、図2に示す実施の形態において、HDD10dに格納されているOSが起動され、そのOS上で、本発明に係るコンパイラ、リンカ、および、ライブラリが実行される際のそれぞれの対応関係を示す図である。この図に示すように、実行対象となるソースプログラム20は、コンパイラ21によってコンパイル（翻訳）され、リンカ22によって基本的なプログラムであるライブラリ23に格納された必要なプログラムが付加され、実行形式プログラム24が生成される。なお、本発明は、ライブラリ23の詳細に係るものであり、以下に示す新たな

制御方法を用いることにより、並列処理の高速化を図ることが可能となる。

【0028】

図4は、図3に示すソースプログラム20の一例を示す図である。

このソースプログラム20では、行番号「1」において、要素数が1000, 2000, 3000である整数型の配列m, n, pが宣言されている。行番号「3」においては並列処理を行うことが宣言されており、これは行番号「19」に示す「END」と対を構成している。行番号「4」から「8」まではループ処理を形成しており、変数iの値を配列mの各要素として格納する処理が実行される。また、行番号「9」から「13」までは他のループ処理を形成しており、変数iを2倍した値を配列nの各要素として格納する処理が実行される。更に、行番号「14」から「18」までは他のループ処理を形成しており、変数iを3倍した値を配列pの各要素として格納する処理が実行される。

【0029】

以下では、行番号「4」から「8」までを第1のループ処理、行番号「9」から「13」までを第2のループ処理、行番号「14」から「18」までを第3のループ処理と呼ぶことにする。行番号「8」, 「13」, 「18」に示す「NOWAIT」は、ウェイト処理を実行しないことを示しており、この一行を付加することにより、バリア機能が停止されることになる。

【0030】

なお、処理の並列数は、図示せぬ初期設定用のプログラムによって指定することができる。また、プログラム中においても、所定のコマンドを配置することにより、並列数を任意に指定することができる。

【0031】

以上のようなソースプログラム20は、コンパイラ21によって翻訳され、リンカ22によってライブラリ23に含まれる必要なプログラムが付加され、実行形式プログラム24に変換される。なお、コンパイルの際には、並列処理の単位である並列処理ブロックに分割され、それぞれの並列処理ブロックにはユニークな番号である並列処理ブロック番号が付与される。図4に示すソースプログラムの例では、第1～第3のループ処理のそれぞれが並列処理ブロックであるので、

例えば、1～3の並列ブロック番号がそれぞれの並列処理ブロックに付与される。

【0032】

図5は、実行形式プログラム24が実行された際の動作を説明するための図である。この図において、実行形式プログラム24は、図4に示すソースプログラム20が機械語に翻訳されたものであり、前述した第1～第3のループ処理を有している。スレッド制御部60および第1～第5のスレッド61～65は、リンカ22によって付加された並列処理用のプログラムであり、実行形式プログラム24から呼び出されて実行されるサブルーチン形式のプログラムである。

【0033】

ここで、スレッド制御部60は、並列処理ブロックである第1～第3のループ処理から呼び出された場合には、第1～第5のスレッド61～65を起動し、処理を実行させるとともに、これらのスレッドの実行状態の管理を行う。第1～第5のスレッドは、PU10a-1～10a-5によって実行される処理の基本単位である。

【0034】

図6は、図5に示すスレッド制御部60および第1～第5のスレッド61～65が機能する際にRAM10cに確保する記憶領域を示す図である。この図において、スレッド情報域71～75は第1～第5のスレッド61～65のそれぞれが確保している領域であり、各情報域にはそのスレッドが現在実行している並列処理ブロックの番号を示す並列処理ブロック番号71a～75aが格納されている。並列処理制御情報域76は、スレッド制御部60が確保している領域であり、全てのスレッドのうち最先のスレッドが実行している並列処理ブロック番号76aが格納されている。

【0035】

並列処理ブロック制御情報域77、78は、現在実行中の並列処理ブロックに対応して生成される情報域であり、当該並列処理ブロックの処理を終了したスレッドの個数である実行終了スレッド数77a、78aが格納されている。この例では、2つの並列処理ブロック制御情報域77、78が生成されており、2つの

並列処理ブロックが実行中の状態に対応している。なお、この例では、第1のスレッド61～第4のスレッド64は、並列処理ブロック制御情報域77に対応する並列処理ブロックを実行中であり、第5のスレッド65は、並列処理ブロック制御情報域78に対応する並列処理ブロックを実行中である。

#### 【0036】

なお、3以上の並列処理ブロックが実行中である場合にはその数に対応した並列処理ブロック制御情報域が生成され、また、1つの並列ブロックが実行中である場合には1つの並列処理ブロック制御情報域が生成される。

#### 【0037】

次に、図7を参照して、図4に示すソースプログラム20から生成された実行形式プログラム24が実行される際の、図5および図6に示すブロック図の動作について説明する。

#### 【0038】

実行形式プログラム24が起動されると、並列処理ブロック番号が「1」である第1のループ処理が実行の対象となる。第1のループ処理は、スレッド制御部60をサブルーチンコールし、引数等を引き渡すとともに、処理の開始を依頼する。すると、スレッド制御部60は、並列処理制御情報域76の並列処理ブロック番号を「1」に設定するとともに、並列処理ブロック制御情報域77を生成し、実行終了スレッド数77aを「0」に初期設定する。続いて、スレッド制御部60は、第1～第5のスレッドを生成して、処理の実行を開始する。第1～第5のスレッド61～65は、それぞれ、スレッド情報域71～75を確保するとともに、現在実行中である第1のループ処理に対応する並列処理ブロック番号「1」を並列処理ブロック番号71a～75aとして格納する。

#### 【0039】

図7は、スレッド情報域71～75と、並列処理制御情報域76に格納されているデータを示す図であり、左端の第1列目から第5列目までがスレッド情報域71～75にそれぞれ対応し、右端の列が並列処理制御情報域76に対応している。この図の第1行目は、処理が開始された当初の状態を示しており、開始当初は第1のスレッド61のみが並列処理ブロック番号「1」を格納しておりその他

はまだ未格納の状態である。第 2 行目では、5 列目以外は全て「1」の状態となっており、第 5 のスレッド 6 5 以外は全て第 1 のループ処理を実行中の状態であることが示されている。

#### 【0040】

このような状態において、第 2 のスレッド 6 2 が第 1 のループ処理の実行を終了すると、スレッド制御部 6 0 にその旨が通知される。すると、スレッド制御部 6 0 は、第 1 の処理ループの末尾に「NOWAIT」が挿入されていることを検出し、バリア機能を実行せずに次の処理に移行することを認識し、先ず、実行終了スレッド数 7 7 a を「0」から「1」に更新する。次に、スレッド制御部 6 0 は、並列処理制御情報域 7 6 の並列処理ブロック番号 7 6 a を「1」から「2」に更新し、続いて、第 2 のループ処理に対応した並列処理ブロック制御情報域 7 8 を生成して実行終了スレッド数 7 8 a を「0」に初期設定する。

#### 【0041】

以上のような動作が繰り返され、全てのスレッドが第 1 のループ処理を終了した場合には、実行終了スレッド数 7 7 a が「5」になるので、その場合には、並列処理ブロック制御情報域 7 7 が RAM 1 0 c 上から削除されることになる。

#### 【0042】

そして、全てのスレッドが第 3 のループ処理を終了した場合には、図 7 の最後の行に示すように、並列処理ブロック番号 7 1 a ～ 7 5 a が全て「3」の状態になり、プログラムの実行が完了することになる。

#### 【0043】

従って、以上の実施の形態では、図 1 0 に示す従来例とは異なり、図 8 に示すようにバリア機能が実行されることなく、各スレッドが他のスレッドの実行状態に拘わりなく、次の並列処理ブロックを実行することになる。従って、図 1 0 に示す従来例では、トータルの実行時間は各並列処理ブロックで最も遅いスレッドの実行時間の合計になるが、本実施の形態ではそれ以下の時間で実行することが可能となる。

#### 【0044】

なお、以上の実施の形態では、5 つの PU 1 0 a - 1 ～ 1 0 a - 5 により、5

つのスレッドで処理する場合を例に挙げて説明したが、本発明はこのような場合に限定されるものではなく、これ以外の組み合わせでも本発明を適用可能であることはいうまでもない。

【 0 0 4 5 】

また、以上の実施の形態では、スレッドの管理をライブラリ 2 3 によって行うようにしたが、コンパイラ 2 1 によって同様のプログラムを実行形式プログラム 2 4 に付加するようにしても同様の効果を得ることができる。なお、付加の方法としては、例えば、インライン展開やマクロ等を用いることができる。

【 0 0 4 6 】

更に、以上の実施の形態では、全ての並列処理ブロックの最後に「NOWAIT」を挿入したので、各スレッドは、他のスレッドとは無関係に最後の並列処理ブロックまで処理を継続することになるが、この「NOWAIT」を除外することにより、そこで、スレッドの足並みを一旦揃えることも可能である。このように「NOWAIT」を適宜挿入することにより、バリア機能を有効または無効にすることが可能となる。

【 0 0 4 7 】

続いて、図 9 を参照し、スレッド制御部 6 0 において実行されるフローチャートについて説明する。このフローチャートは、第 1 ～ 第 5 のスレッド 6 1 ～ 6 5 において処理が終了した場合に呼び出されて実行される。このフローチャートが開始されると、以下のステップが実行される。

【 0 0 4 8 】

ステップ S 1 0 :

該当するスレッド情報域の並列処理ブロック番号を 1 だけインクリメントする。

【 0 0 4 9 】

ステップ S 1 1 :

変数 j に並列処理制御情報域の並列処理ブロック番号を代入する。

ステップ S 1 2 :

変数 k に該当するスレッド情報域の並列処理ブロック番号を代入する。



【0050】

ステップS13：

変数jの値が変数kの値以上であるか否か、即ち、当該スレッドが新たな並列処理ブロックを実行するか否かを判定し、新たな並列処理ブロックを実行する場合にはステップS14に進み、それ以外の場合にはステップS17に進む。

【0051】

ステップS14：

並列処理制御情報域76の並列処理ブロック番号76aを1だけインクリメントする。

【0052】

ステップS15：

新たな並列処理ブロックに対応する並列処理ブロック制御情報域を生成するとともに、実行終了スレッド数を「0」に初期設定する。

【0053】

ステップS16：

新たなスレッドの実行処理を開始する。

ステップS17：

スレッド情報域の並列処理ブロック番号を参照し、実行対象となる並列処理ブロックを特定する。

【0054】

ステップS18：

ステップS17で特定した並列処理ブロックを実行する。

ステップS19：

該当する並列処理ブロック制御情報域の実行終了スレッド数をインクリメントする。

【0055】

ステップS20：

ステップS19におけるインクリメントの結果、実行終了スレッド数がスレッド数（図6の例では「5」）と等しくなった場合には、ステップS21に進み、

それ以外の場合には処理を終了する。

【 0 0 5 6 】

ステップ S 2 1 :

処理が終了した並列処理ブロック制御情報域を削除する。

以上の処理によれば、前述した機能を実現することが可能となる。

【 0 0 5 7 】

最後に、上記の処理機能は、コンピュータによって実現することができる。その場合、情報処理装置が有すべき機能の処理内容は、コンピュータで読み取り可能な記録媒体に記録されたプログラムに記述されており、このプログラムをコンピュータで実行することにより、上記処理がコンピュータで実現される。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場へ流通させる場合には、C D - R O M (Compact Disk Read Only Memory) やフロッピーディスク等の可搬型記録媒体にプログラムを格納して流通させたり、ネットワークを介して接続されたコンピュータの記憶装置に格納しておき、ネットワークを通じて他のコンピュータに転送することもできる。コンピュータで実行する際には、コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行する。

【 0 0 5 8 】

(付記 1) 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させる情報処理方法において、

実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割ステップと、

前記並列処理ブロック分割ステップによって分割された並列処理ブロックを、前記複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割ステップと、

所定のプロセッサにおいて、前記スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示ステップと、

を有することを特徴とする情報処理方法。

【 0 0 5 9 】

(付記 2) 前記指示ステップは、前記実行対象となるプログラムに所定の指示がなされている場合には、全てのスレッドの処理が終了するまで、次の並列処理ブロックの実行を指示しないことを特徴とする付記 1 記載の情報処理方法。

【0060】

(付記 3) 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において

コンピュータを、

実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割手段、

前記並列処理ブロック分割手段によって分割された並列処理ブロックを、前記複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割手段、

所定のプロセッサにおいて、前記スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示手段、

として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体

。

【0061】

(付記 4) 複数のプロセッサを有し、所定の情報処理を実行する情報処理装置において、

実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割手段と、

前記並列処理ブロック分割手段によって分割された並列処理ブロックを、前記複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割手段と、

所定のプロセッサにおいて、前記スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示手段と、

を有することを特徴とする情報処理装置。

【0062】

(付記 5) 複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において

コンピュータを、

複数の並列処理ブロックに分割された実行対象のプログラムの所定の並列処理ブロックから実行要求がなされた場合には、複数のスレッドを生成し、各プロセッサに処理を分担させる処理分担手段、

何れかのスレッドの処理が終了した場合には、次の並列処理ブロックに係るスレッドの実行を指示する実行指示手段、

として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体

【 0 0 6 3 】

(付記 6) 前記実行指示手段は、前記実行対象となるプログラムに所定の指示がなされている場合には、全てのスレッドの処理が終了するまで、次の並列処理ブロックの実行を指示しないことを特徴とする付記 5 記載の記録媒体。

【 0 0 6 4 】

【発明の効果】

以上説明したように本発明では、複数のプロセッサを有する計算装置に対して、所定の情報処理を実行させる情報処理方法において、実行対象となるプログラムを複数の並列処理ブロックに分割する並列処理ブロック分割ステップと、並列処理ブロック分割ステップによって分割された並列処理ブロックを、複数のプロセッサのそれぞれに分担して処理させるための基本処理単位であるスレッドに分割するスレッド分割ステップと、所定のプロセッサにおいて、スレッドの実行が終了した場合には、次の並列処理ブロックの実行を指示する指示ステップと、を設けるようにしたので、並列処理ブロックが連続して処理される場合には、従来のようなバリアによる遅延を排除することにより、処理速度を向上させることが可能となる。

【図面の簡単な説明】

【図 1】

本発明の動作原理を説明する原理図である。

【図 2】

本発明の実施の形態の構成例を示すブロック図である。

【図 3】

図 2 に示す実施の形態において、HDD に格納されている OS が起動され、その OS 上で、本発明に係るコンパイラ、リンカ、および、ライブラリが実行される際のそれぞれの対応関係を示す図である。

【図 4】

図 3 に示すソースプログラムの一例を示す図である。

【図 5】

実行形式プログラムが実行された際の動作を説明するための図である。

【図 6】

図 5 に示すスレッド制御部および第 1 ～ 第 5 のスレッドが機能する際に RAM に確保している記憶領域を示す図である。

【図 7】

図 4 に示すソースプログラムから生成された実行形式プログラムが実行される際の、図 5 および図 6 に示すブロック図の動作について説明する図である。

【図 8】

本実施の形態におけるスレッドの実行形態を説明するための図である。

【図 9】

図 5 に示すスレッド制御部において実行される処理の一例を説明するフローチャートである。

【図 10】

従来例におけるスレッドの実行形態を説明するための図である。

【符号の説明】

- 1 並列処理ブロック分割ステップ
- 2 スレッド分割ステップ
- 3 指示ステップ
- 4 プロセッサ群

10a-1~10a-5 PU

10b ROM

10c RAM

10d HDD

10e GB

10f I/F

10g バス

11 表示装置

12 入力装置

20 ソースプログラム

21 コンパイラ

22 リンカ

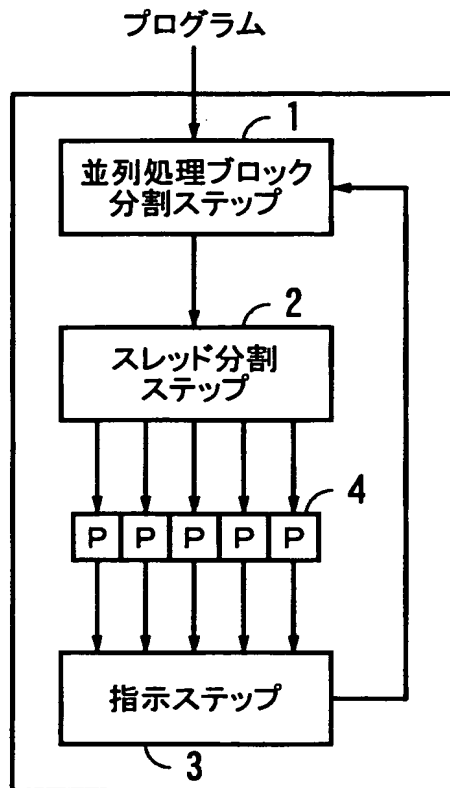
23 ライブラリ

24 実行形式プログラム

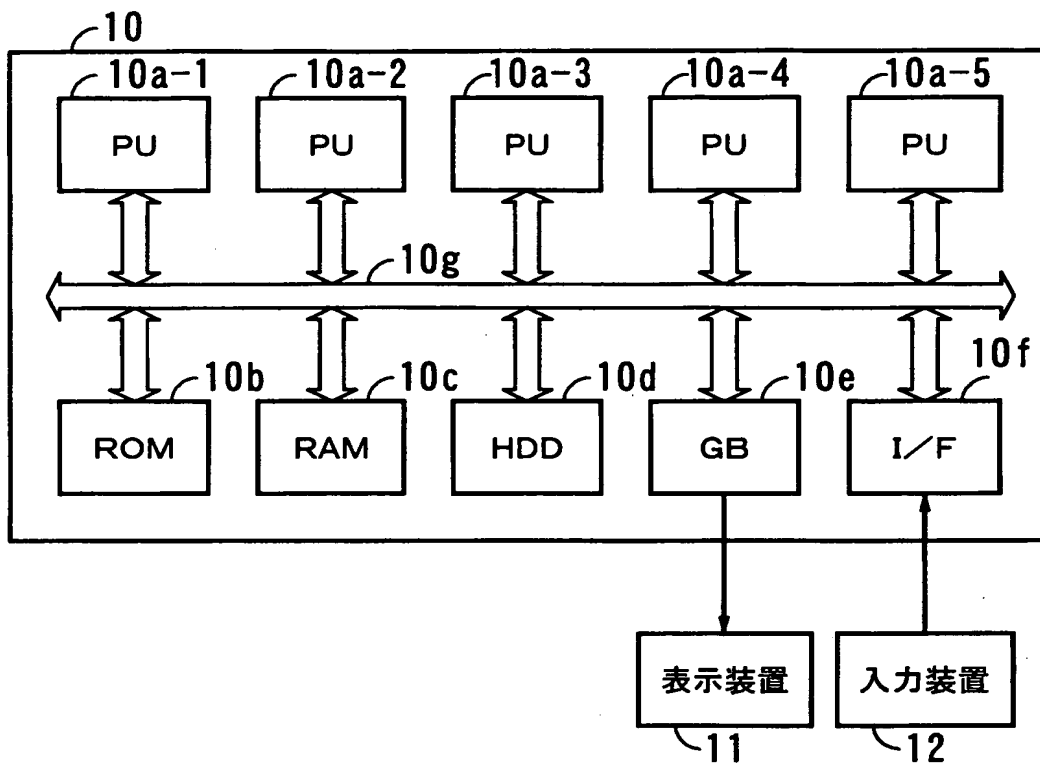
【書類名】

図面

【図 1】

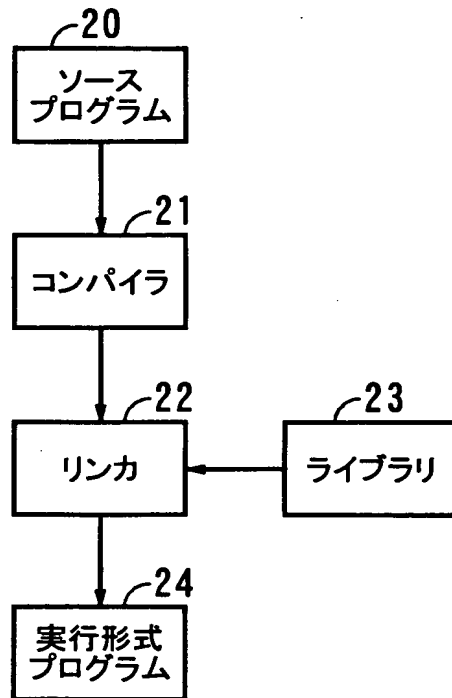


【図2】





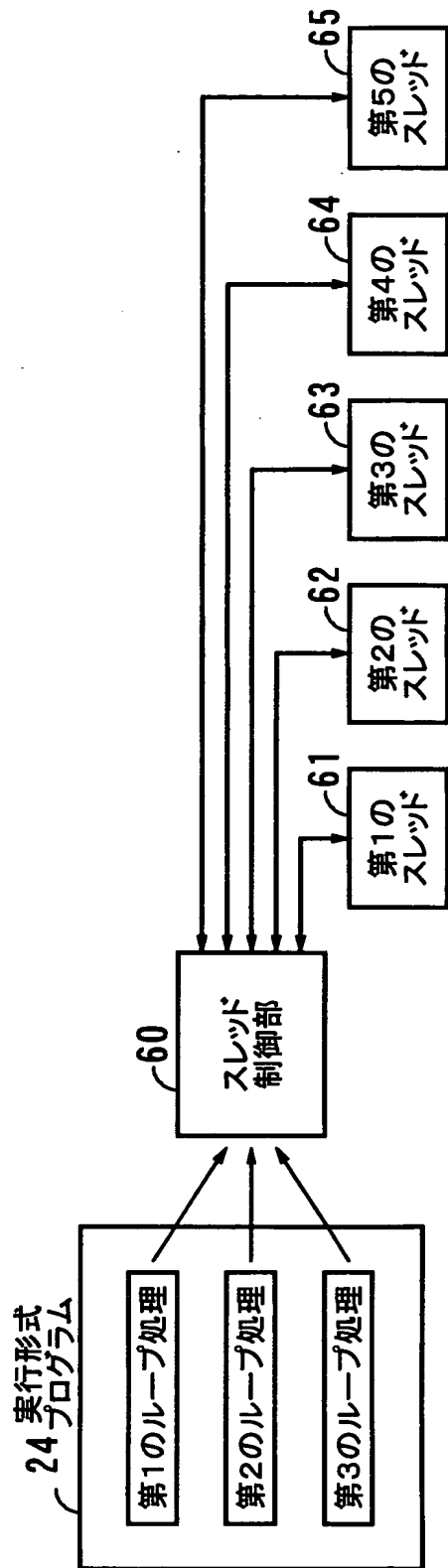
【図 3】



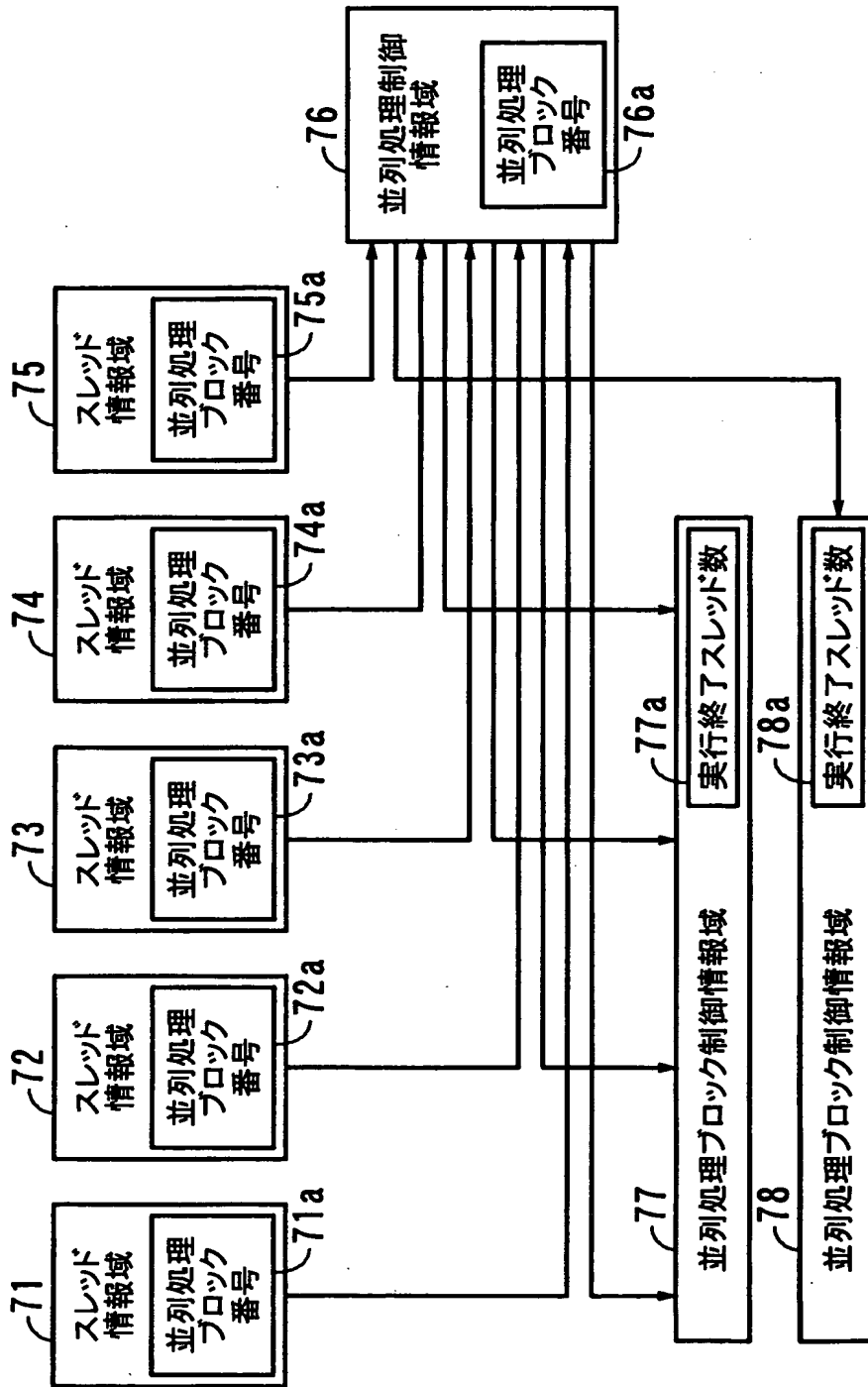
【図 4】

```
1  INTEGER m(1000), n(2000), p(3000)
2
3  !$OMP PARALLEL
4      !$OMP DO
5          DO i=1, 1000
6              m(i)=i
7          END DO
8      !$OMP END DO NOWAIT
9      !$OMP DO
10         DO i=1, 2000
11             n(i)=i*2
12         END DO
13     !$OMP END DO NOWAIT
14     !$OMP DO
15         DO i=1, 3000
16             p(i)=i*3
17         END DO
18     !$OMP END DO NOWAIT
19 !$OMP END PARALLEL
20 END
```

【図 5】



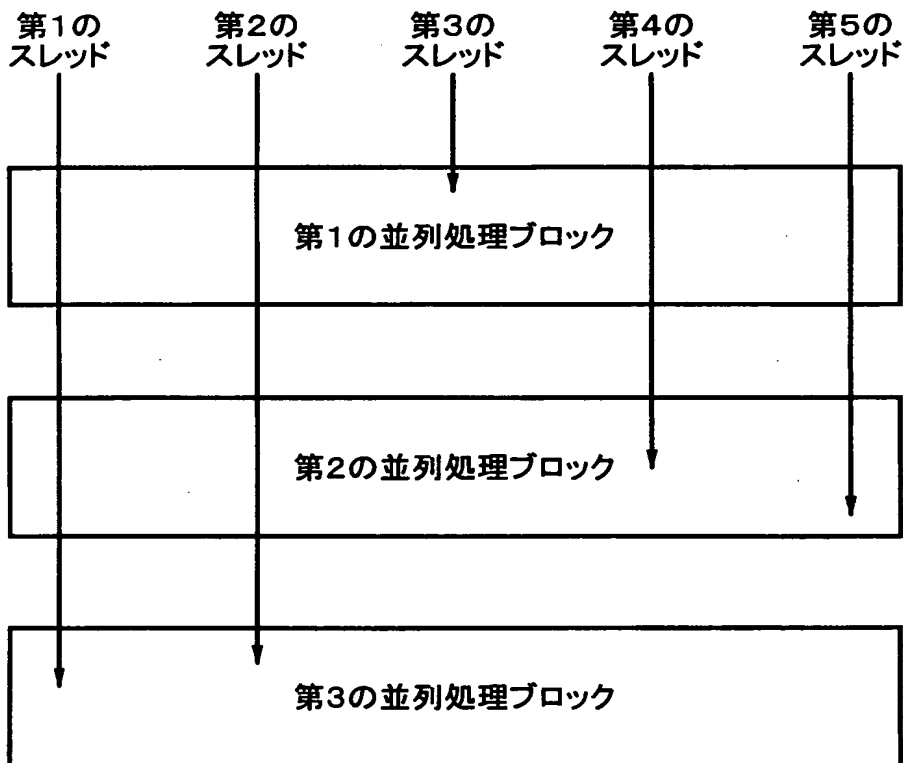
【図 6】



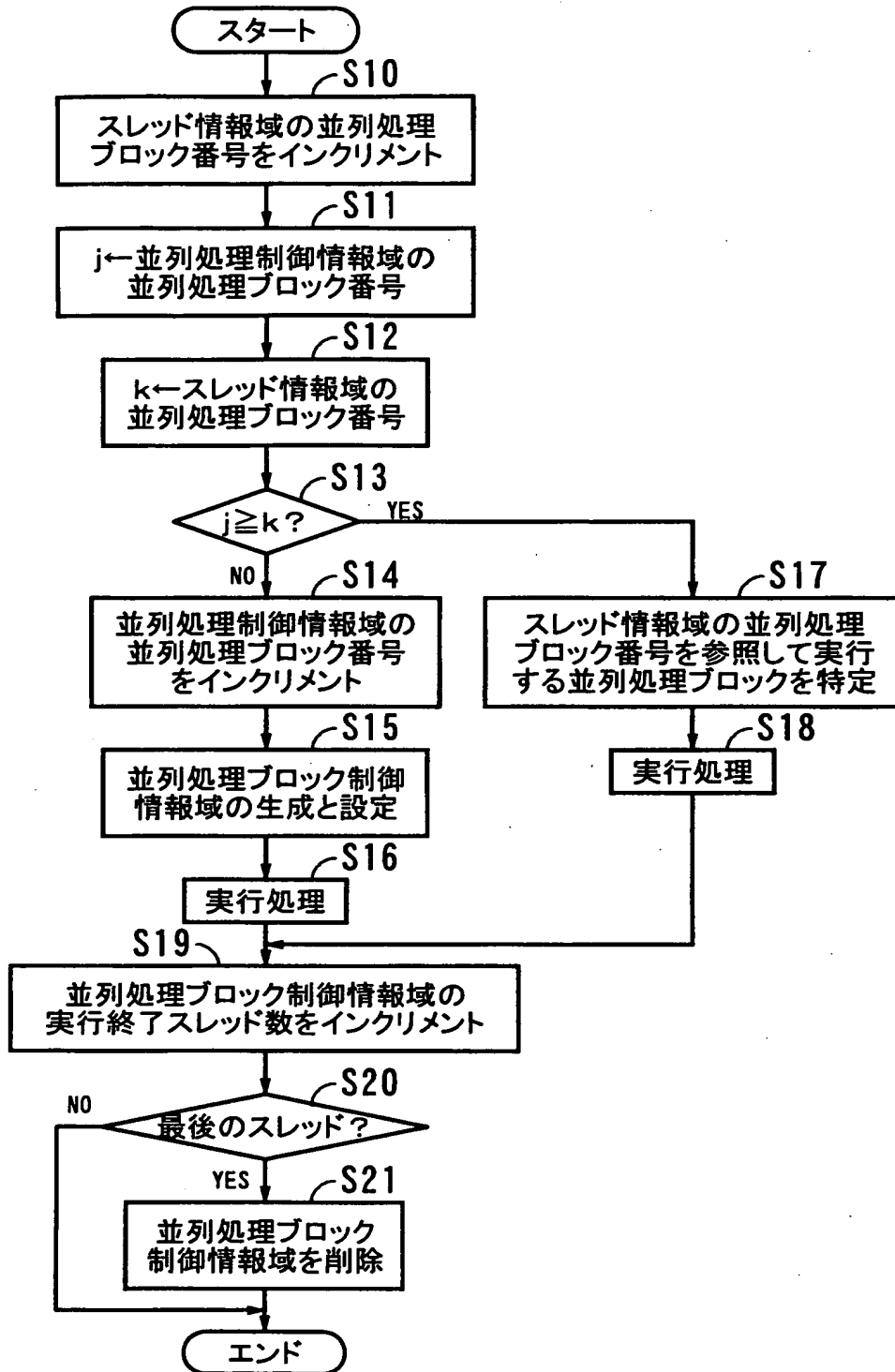
【図 7】

第1の スレッド	第2の スレッド	第3の スレッド	第4の スレッド	第5の スレッド	先頭
1	-	-	-	-	1
1	1	1	1	-	1
1	2	1	1	1	2
⋮	⋮	⋮	⋮	⋮	⋮
3	2	2	2	2	3
3	3	3	3	3	3

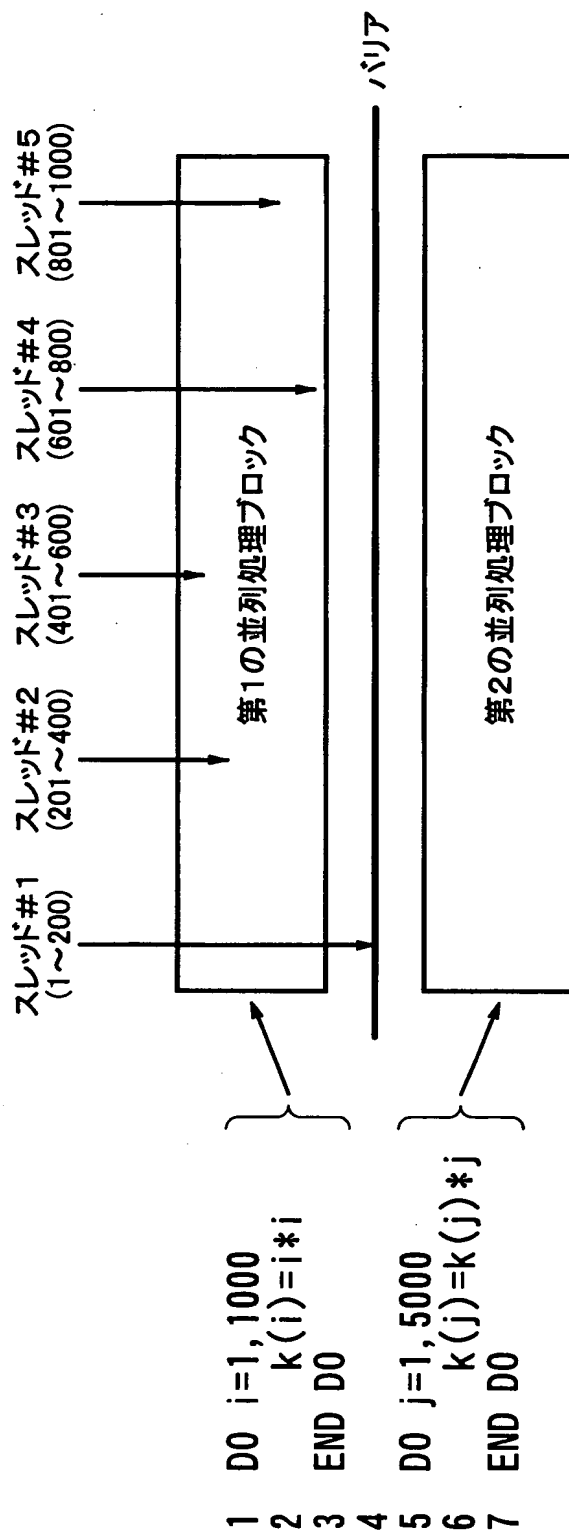
【図 8】



【図 9】



【図10】



【書類名】            要約書

【要約】

【課題】    並列処理計算装置の実行速度を向上させる。

【解決手段】    並列処理ブロック分割ステップ1は、実行対象となるプログラムを複数の並列処理ブロックに分割する。スレッド分割ステップ2は、並列処理ブロック分割ステップ1によって生成された並列処理ブロックを、プロセッサ群4の個数に応じた複数のスレッドに分割する。プロセッサ群4は、スレッド分割ステップ2によって生成されたスレッドを実行する。指示ステップ3は、プロセッサ群4のそれぞれのプロセッサがスレッドの実行を終了した場合であって、プログラムに所定の指示がなされていない場合には、当該プロセッサに対して次の並列処理ブロックに係るスレッドを実行するように指示する。

【選択図】            図1



出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社